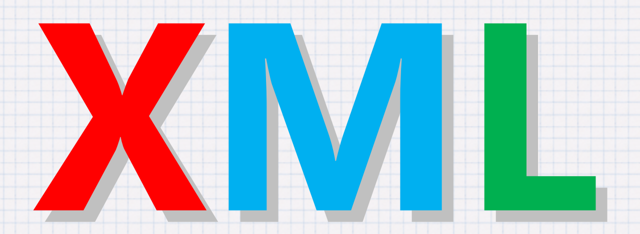
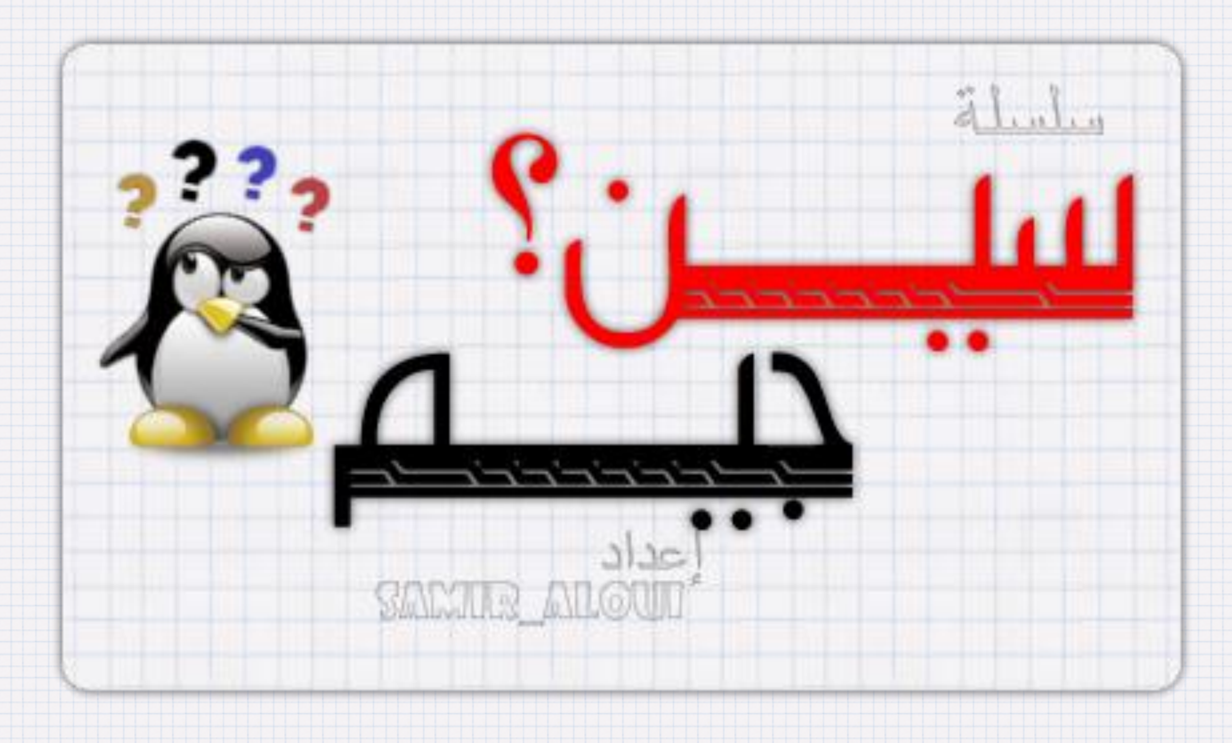
موضع اليوم حول:



الجزء الأول



بسم الله الرحمن الرحيم

و الصلاة و السلام على أشرف المرسلين سيدنا محمد خاتم الأنبياء و المرسلين

أما بعد

س: ما هي "سين جيم"؟

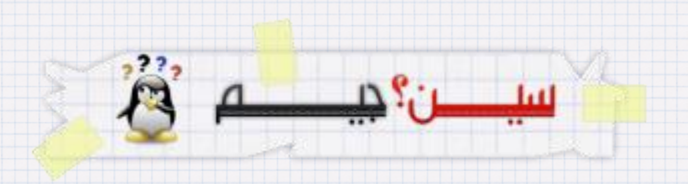
ج: "سين جيم" هي سلسلة من المواضيع الخفيفة و الهامة لكل مبرمج. إذ أننا في كل مرة سنتناول فكرة معينة، مبدأ معين، تعريف معين، و يكون من الأمور التي تصادفنا كمبرمجين مع محاولة التركيز على الشرح و مبدأ العمل و خطوات الاستعمال بالترتيب في حين سنحاول قدر الإمكان استعمال الخطوات بدل الأكواد لتفادي التعارض بين اللغات. أما في حالة كانت هناك ضرورة تستدعي استخدام الأكواد البرمجية للغة معينة فسنضعها للإيضاح فقط.

أتمنى أن تنال السلسلة رضاكم و أرجو أن لا تبخلوا علي بأي سؤال يعتريكم في أمر من أمور البرمجة كي أضعه في موضوع بإذن الله ليستفيد الجميع. و أجري و أجركم على الله و الله من وراء المقصد.

أخوكم سمير علوي

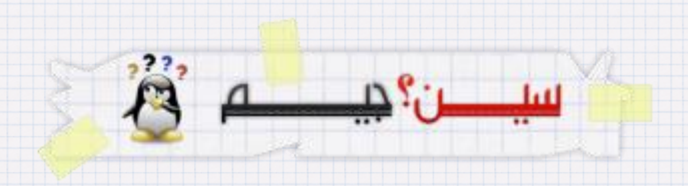
س: ما هو موضوع اليوم؟

ج: في موضوع اليوم سأحاول التطرق فيه إلى مفهوم جديد و لكنه لاقى انتشارا كبيرا في كل لغات البرمجة الحديثة. هذا المفهوم هو الـ XML. ولما صار من الضروري معرفته بشكل يزيل الغموض و يوضح كيف يبدو و كيف يستعمل. إذن لنبدأ على بركة الله



س: ما معنى XML؟

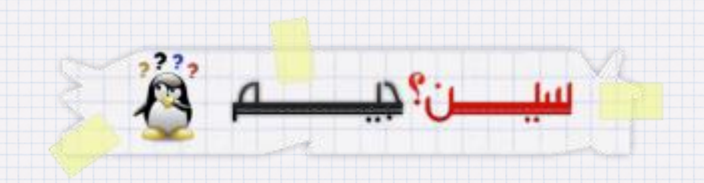
ج: XML هو اختصار للجملة eXtended Markup Language. تم اشتقاقه من HTML التي تعني HyperText Markup Language.



س: ما هي أوجه الشبه بين XML و HTML؟

ج: XML تم اشتقاقه من HTML فهو يكتب بنفس الشكل تقريبا إذ لديه وسوم أو ما يعرف بـ Tag كل وسم يكون بالشكل <TagName params>Contenu</TagName> حيث تعتبر الأولى وسم الفتح و الأخيرة التي بها رمز / يعتبر وسما لإغلاق و بينهما المحتوى الذي سيقع عليه تأثير الوسم. يمكن إضافة بارامترات لوسم الفتح تماما مثل ما نفعله فيHTML كما أيضا يوجد ما يسمى بالوسوم الفارغة فمثل
 يوجد أيضا <TagName/>

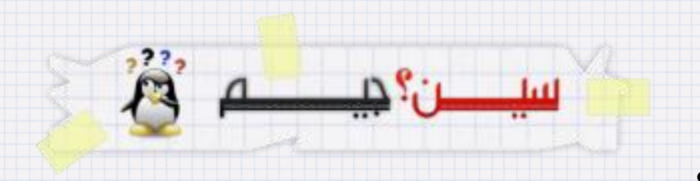
نقطة أخرى و هي كون كل من ملف XML و HTML يكون على شكل شجري و يكون لديه فرع واحد رئيسي لا أكثر. ففي HTML نجد الوسم الرئيسي هو <html></html> و كل شيء يضاف يكون داخله لا خارجه. و الأمر نفسه في ملف XML يجب أن يكون هناك فرع رئيسي واحد و وحيد و كل الوسوم الأخرى تكون إما أبناءه المباشرين أو أبناءه غير المباشرين.



س: ما هي أوجه الاختلاف بين XML و HTML؟

ج: هناك أربعة اختلافات جوهرية بين XML و HTML هي:

- أول وجه للاختلاف هو كون الوسوم الخاصة بـ HTML معروفة و محددة سابقا و كذلك أنواع قيمها و أسماء البارامترات الخاصة بكل وسم. أما XML فلكل شخص أن يضع الوسوم التي يريد بالقيم التي يريد و بأي عدد و اسم من البارامترات يريد. و كل ما عليه هو التقيد بالشكل العام فقط.
- ثاني اختلاف في التراكب ففي HTML العملية التالية ممكنة نفتح وسم A...نفتح وسم B...نغلق الوسم A...نغلق الوسم B بينما هذه العملية مستحيلة في XML إذ لا يمكن غلق وسم A حتى تغلق الوسوم التي فتحت بعده بالترتيب أي أن آخر وسم فتح هو أول وسم يغلق. فالأمر أشبه بحلقات لغات البرمجة لو بدأت حلقة A ثم بدأت حلقة B ثم بدأت حلقة C فالإغلاق يكون بالشكل C ثم B ثم A و إلا ستحصل على خطأ.
- ثالث اختلاف أن HTML له محركات تنفذه ليظهر لنا صفحات ويب. بينما XML ليس له محركات خاصة. و لكن يمكن بناءها له حسب الحاجة و هناك برامج تقوم بذلك مثل SpyXML.
- و آخر اختلاف هو أن قيم HTML تكون من نوعين فقط: إما نصية أو رقمية. بينما القيم في XML يمكن أن تكون بيتات أيضا و حين نقول بيتات فمعناها كائنات مثل : صورة، صوت، ملف تنفيذي،...إلخ.



•

س: في ماذا يستعمل XML؟

ج: وظيفة XML الأساسية هي تخزين البيانات مثله مثل HTML فهو يحفظ قيما معينة. أما مجالات استعماله فهي عكس HTML الذي يستعمل في إظهار محتوى ما على متصفح ويب. و يتوقف استعمال ملف XML حسب حاجتنا و حسب ترجمتنا لقيمه و من بين أهم الاستعمالات الحالية:

- يستعمل كبديل لملف حفظ الإعدادات ذي الامتداد .INI الذي كان يستخدم سابقا في حفظ قيم الإعدادات الخاصة بالتطبيق. مثل لغة الواجهة و الألوان و الستايل المستعمل و إعدادات الشاشة (للألعاب) و ... إلخ.
- يستعمل في نقل البيانات من قاعدة لأخرى حين لا يكون هناك تماثل في قواعد البيانات من حيث النوع (Oracle مع SQL Server مثلا) و لا يكون هناك توافق في أسماء الحقول (في الجدول الأول نجد: Nom, Prenom و في الجدول الثاني: FirstName, LastName) و لا يكون هناك توافق في نوع البيانات (في الجدول الأول العمود A من نوع Integer و في الجدول B من نوع Double) و لا يكون هناك توافق في عدد الأعمدة (احد الجدولين به أعمدة إضافية لا توجد في الجدول الآخر).
- يستعمل في تخزين قيم متغيرات و جداول و حتى كائنات بكاملها ثم يتم استرجاعها لاحقا بقيمها و حالتها التي حفظت بها. لنكمل التنفيذ من تلك النقطة التي توقفنا عندها يوم حفظ تلك القيم و تلك الكائنات.
- يستعمل كملف لتخزين الواجهات Skin أو اللغات Language أو المظهر الخارجي لـجداول البيانات DataSet و DataTabel..إلخ.
- يستعمل في حفظ لغة البرمجة الوسيطة مما يمكن لاحقا من برمجة محركات تعرف خاصة تحول اللغة الوسيطة إلى أي لغة نريدها و ابسط مثال MSIL الخاصة بمايكروسوفت للغات الـ DotNet.
- تستخدم في تخزين مشروع كاملا بكل ملفاته و مجلداته و مصادره بدل تخزينه بشكل مجلد بداخله ملفات و ما ينجم عن ذلك من احتمال تعطل البرنامج في حال ضياع أحدها أو تغيير مكانه على الجهاز. و هذه الأخيرة لي فيها مشروع مشترك مع الأخ عبد القادر في طريق الانجاز.
- يتم توليد لغات برمجة وصفية جديدة و قوية منها مثل XAML اللغة المسؤولة عن تصميم الواجهات الرسومية في برامج WPF التي ظهرت مع المنصة Framwork 3.5.

هناك استعمالات أخرى لا حصر لها لملفاتXML تختلف حسب اختلاف مستعملها و طريقة تعامله مع بياناتها.

س: ما هو تركيب الملف XML؟

ج: يتكون ملف XML من:

:XMLDocument

و هو أهم شيء في ملف XML و بدونه لا يمكن تعريف كائن XML رغم أن هذا الجزء مجرد مفهوم و ليس له شكل و لا أي شيء يتم تعريفه و لكنه يعتبر هو الحامل للفرع الرئيسي. فلو شبهنا الأمر بشجرة حقيقية مثلا فهو يعتبر التربة التي تمسك الجذع الرئيسي و من دونها لا يكون لدينا شجرة واقفة.

:XMLNode

كل الأنواع التي ستأتي فيما بعد مشتقة من XmlNode حيث يعتبر هذا العنصر هو الشكل العام لأي عنصر.

:XMLDeclaration

هو عنصر ليس أساسيا لكنه يكون ضروريا أحيانا و ظيفته تعريف نسخة XML المستعملة و كذا نوع الـEncoding و طبعا هذا سيمكن مستعمل الملف لاحقا من معرفة هل يمكنه قراءة الملف أم لا و هل القيم ستؤخذ مثلا كـ Ascii أم Unicode أم utf-8 أم أي نوع من الـ Encoding.

:XMLElement

بصفة عامة هو كل وسم داخل ملف XML في المثال التالي: < tag1 param1="10">Hello world!</tag1> نقول أن هذا الـ XmlElement يدعى tag1. و يعتبر الفرع الرئيسي لأي ملف XML من نوع XMLElement.

:XMLText

هو نوع خاص من XMLNode يتمثل في القيمة التي يأخذها XMLElement فمن المثال السابق "Hello ! world" هي قيمة الـ XMLTextالمرتبط مع الـ XMLElement المدعو "tag1".

:XMLAttribut

نوع مشتق من XMLNode و هو يمثل بارامترا مع قيمته. من المثال السابق فإن الفرع tag1 يملك بارامترا اسمه param1 و قيمة هذا البارامتر هي 10. مع مراعاة أن القيمة دائما تكتب بين " " سواء كانت قيمته

نصية أو عددية أو منطقية أو أي قيمة كانت. بالإضافة إلى هذا توجد بعض الأنواع الأخرى المشتقة من XMLNode منها XmlComment التي تضع وسم تعليق. و XmlEntity التي تضع رأسية للملف و Xml.

س: كيف ننشئ ملف XML اعتمادا على العناصر المذكورة أعلاه؟

ج: لصناعة ملف XML بسيط نتبع الخطوات التالية بالترتيب:

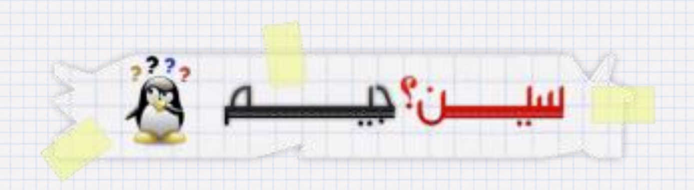
1. نبدأ أولا بتعريف أهم شيء و هو XmlDocument و ذلك باستعمال دالة بناء بالشكل New XmlDocument و ليكن اسمه xmlDoc. و يعتبر هذا هو العنصر الوحيد الذي يتم إنشاؤه باستعمال دالة بناء New أما بقية العناصر فكلها يتم إنشاؤها من طرف العنصر xmlDoc بواسطة دوال خاصة و بسيطة. و هنا أريد وضع ملاحظة مهمة جدا و هي أن إنشاء أي عنصر لا يجعله يعني أنه تم إضافته. بل يجب ربطه باستعمال دالة خاصة تدعى AppendChild الخاصة بالعنصر الذي نريد أن نجعل العنصر الجديد ابنا له.
2. الآن نقوم بإنشاء عنصر من نوع XmlDeclaration لإضافة التعريف و ذلك باستعمال الدالة xmlDoc.CreateXmlDeclaration و نمرر لها كباراميترات: رقم النسخة، و نوع الـ Encoding. و بما أن هذا العنصر يكون ابنا لـ xmlDoc فإننا نضيفه باستعمال الدالة xmlDoc.AppendChild و نمرر لها هذا العنصر.
3. نقوم الآن بتعريف ثاني أهم عنصر ألا و هو الفرع الرئيسي الذي يسكون أبا لكل الفروع الأخرى. كما قلنا فالإنشاء من وظيفة xmlDoc باستعمال الدالة CreateElement و نمرر لها إسم العنصر. ثم نقوم بعدها بربطه بالـ xmlDoc بالدالة xmlDoc.AppendChild و ليكن إسمه xmlRoot
4. يمكننا الآن إنشاء ما نريد من XmlElement بواسطة الدالة xmlDoc.CreateElement مع تمرير الاسم لها و لا بأس لو كان هناك نفس الاسم متكررا. أما عن الربط فلا يمكن الربط مع xmlDoc بل يجب الربط إما بـ xmlRoot باستعمال الدالة xmlRoot.AppendChild أو نربطها بأي XmlElement آخر و ليكن اسمه xmlParent بالشكل: (xmlParent.AppendChild (xmlChild حيث xmlChild هو العنصر الذي أنشأناه حديثا.
5. إذا كان XmlElement و ليكن اسمه eX يملك باراميتر يدعى param و قيمته 10 مثلا فإنه توجد طريقتان لتعريفه
   a. الطريقة الأولى: و هي الطريقة المباشرة: حيث نستعمل مباشرة الدالة eX.SetAttribute("param",10")
   b. الطريقة الثانية و هي الطريقة غير المباشرة و تتم وفق المراحل التالية:
      i. نقوم بإنشاء xmlAttribute باستعمال الدالة("xmlDoc.CreateAttribute("param و ليكن إسمه xmlAttrib

ii. نعطيها الآن قيمة بالشكل xmlAttrib.Value=10

iii. نربط الآن البارميتر بالعنصر eX بالشكل eX.SetAttributeNode(xmlAttrib)

كملاحظة فقط فإنه يمكننا تعريف العدد الذي نريده من البارميترات لأي xmlElement و لكن لا يمكن أن يكون لها نفس الاسم و لو أضفنا بارميترات لها نفس الاسم لنفس العنصر فإن اسم البارميتر سيظهر مرة واحد فقط أما القيمة فستكون قيمة آخر بارميتر له نفس الاسم تم إضافته. و لنشرح ببساطة نقول أنه في كلتا حالتي الإضافة فإن الطريقتان للإضافة إن لم يكن البارميتر موجود من قبل أما إن كان موجود مم قبل فإن عملهما هو تغيير القيمة.

6. إذا لم يكن ex يحتوي على قيمة فسيظهر بالشكل التالي <tagName/> أما إن كانت له قيمة فسيظهر بالشكل التالي <tageName>value<tagName/>. و لتعريف قيمة فإنه يجب تعريف عنصر من نوع XmlText باستعمال الدالة xmlDoc.CreatText و نمرر لها القيمة التي نريدها ثم نربط هذه الـ xmlText مع الـ xmlElement المسمى ex باستعمال الدالة ex.AppendChild و نمرر لها الـ xmlText الذي أنشأناه للتو. و هنا أنوه أنه لو أضفنا أكثر من قيمة لأي عنصر فإن القيم ستجمع كلها في جملة واحدة دون فواصل و لا مسافات. لذا في حالة أردنا أن نضيف أكثر من قيمة أن نجمعها كلها قبل وضعها أو أن نقوم بقراءة القيمة السابقة فإن كانت موجودة نضيف رمز فصل مثل ; مثلا ثم نضيف القيمة الجديدة.



س: هل هناك دالة أخرى لإنشاء أي من العناصر السابقة؟

ج: في VS.Net يمكن إنشاء أي عنصر باستعمال الدوال المذكورة السابقة. كما يمكن استعمال دالة أخرى و هي xmlDoc.CreateNode و نمرر لها كبارميترات: نوع الـ Node (نجده في التعداد xmlNodeType) بالإضافة إلى الاسم الذي نريد إعطاءه له.

س: كيف يمكن إنشاء المثال السابق بإتباع الخطوات المذكورة سابقا؟

ج: سأستعمل كود VB.Net بإتباع الخطوات السابقة فنحصل على الكود التالي

```
Imports System.Xml

Private Sub Example()

    'Create a document
    Dim xmlDoc As New XmlDocument()

    'Create a declaration and add it to xmlDoc
    Dim Declaration As XmlDeclaration = xmlDoc.CreateXmlDeclaration("1.0","utf-8",Nothing)
    xmlDoc.AppendChild(Declaration)

    'Create a comment and add it to xmlDoc
    Dim Comment As XmlComment = xmlDoc.CreateComment("tis is a comment line")
    xmlDoc.AppendChild(Comment)

    'Create the main Element (Root) <tag1></Tag1> and add it to xmlDoc
    Dim Root As XmlElement = xmlDoc.CreateElement("tag1")
    xmlDoc.AppendChild(Root)

    Create an attribute "param1" with value=10 and add it to "Root"   <Tag1 param1="10">
    Dim p As XmlAttribute = xmlDoc.CreateAttribute("param1")
    p.Value = 10
    Root.SetAttributeNode(p)

    'Create a direct attribute "param2" value="abcd" from "Root"
    Root.SetAttribute("param2", "abcd")

    'Create a value  "Hello Word!" and add it to "Root"
    Dim Value As XmlText = xmlDoc.CreateTextNode("Hello World!")
    Root.AppendChild(Value)

    'Create a value "Hello too" and add it to "Root"
    Dim Value2 As XmlText = xmlDoc.CreateTextNode("Hello too")
    Root.AppendChild(Value2)

    'Save the document
    xmlDoc.Save("d:\sample.xml")
End Sub
```

و النتيجة بعد التنفيذ تكون بالشكل التالي:

```
 <?xml version="1.0" encoding="UTF-8"?>
<!--tis is a comment line-->
<tag1 param2="abcd" param1="10">Hello World!Hello too</tag1>
```

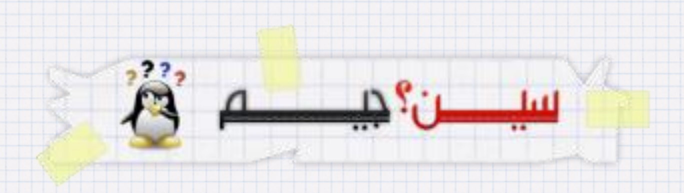
س: هل نستطيع اضافة عنصر في الوسط أو في البداية أو تغيير مكان عنصر ما ؟؟

ج: بالطبع يمكن ذلك بسهولة حيث أنه بدل استعمال الدالة AppendChild فإننا نستعمل إحدى الدالتين InsertAfter أو InsertBefore حيث نمرر لها العنصر الجديد و العنصر الذي سنضع بعده أو قبله. بشرط أن يكون لهما نفس الأب مباشرة. أما بالنسبة لتغيير المكان فإنه يكفي استعمال إحدى الدالتين السابقتين أيضا لأن Insert تضيف العنصر إن لم يكن موجودا و لكن لو كان موجودا فهي تقوم بتغيير مكانه حسب الاسم Before

أو After. ويمكن أيضا حذف أي عنصر باستعمال الدالة RemoveChild. و RemoveAttribute و دوال أخرى تشبهها أتركها لمن يريد البحث عنها في VS.Net

س: هل هناك دوال أخرى؟

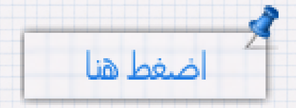
ج:بالطبع هناك دوال أخرى مثل الدالة LoadXml و LoadDocument و ReadNode و غيرها كثير مما لن نتحدث عنه بل ندعه للمرة القادمة



قبل الختام أريد أن أقدم تمرينا بسيطا لمن يريد التطبيق و هو محاولة إنشاء كود يعطينا الملف التالي:

```
<?xml version="1.0" encoding="utf-8"?>
<!--tis sample will create a sample html with XML-->
<html>
  <title>HTML by XML</title>
  <Body text="gold" bgcolor="black">
    <table width="80%" border="1">
      <tr>
        <th>
          <h2>Definition1</h2>
        </th>
        <th>
          this is a sample th
        </th>
      </tr>
      <tr>
        <td rowspan="2">
          <p align="right">we can create an HTML document with XML tags</p>
        </td>
      </tr>
      </table>
    </Body>
</html>
```

للمزيد من التوسع و التفاعل هذا رابط على منتدى لغات البرمجة –منتديات ستارتايمز-



إلى هنا نأتي لنهاية أول موضوع من السلسلة . في انتظار ملاحظاتكم، مشاركاتكم و انتقاداتكم استودعكم الله الذي لا تضيع ودائعه و السلام عليكم و رحمة الله و بركاته